AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Canceled)

2. (Previously Presented) A computer implemented method of identifying an object to be processed by one or more threads of execution comprising:

associating an output of a transport object with an input of an object;

propagating information from an input of the transport object to an output of the transport object; and

propagating the information from the output of the transport object to the object.

3. (Previously Presented) A computer implemented method of identifying an object to be processed by one or more threads of execution comprising:

associating an output of a transport object with an input of an object;

propagating at least one of information concerning, dataset type, information rate or action latency, from an input of the transport object to an output of the transport object; and

propagating the information from the output of the transport object to the object.

4. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an equivalent function descriptor object in place of the function descriptor object based upon the propagated parameter information.

5. (Previously Presented) The computer implemented method of claim 4,

wherein the propagated parameter information includes at least one of dataset type, information rate or action latency.

6. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein for any given set or pattern of input information atoms, the function descriptor object will produce the same set or pattern of output information atoms as the other function descriptor object.

7. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein the function descriptor object is logically equivalent to the other function descriptor object.

8. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein a data set of the function descriptor object is equal to a data set of the other function descriptor object.

9. (Previously Presented) The computer implemented method of claim 8,

wherein the propagated parameter information includes dataset type information.

10. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein an information rate of the function descriptor object is less than or equal to an information rate of the other function descriptor object.

11. (Previously Presented) The computer implemented method of claim 10,

wherein the propagated parameter information includes information rate information.

12. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein an action latency of the function descriptor object is greater than or equal to action latency of the other function descriptor object.

13. (Previously Presented) The computer implemented method of claim 12,

wherein the propagated parameter information includes action latency information.

14. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein the function descriptor object is logically equivalent to the other function descriptor object; and

wherein a data set of the function descriptor object is equal to a data set of the other function descriptor object.

15. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein the function descriptor object is logically equivalent to the other function descriptor object; and

wherein at least one of the following is true,

a data set of the function descriptor object is equal to a data set of the other function descriptor object, or

an information rate of the function descriptor object is less than or equal to an information rate of the other function descriptor object, or

an action latency of the function descriptor object is greater than or equal to action latency of the other function descriptor object.

16. (Previously Presented) The computer implemented method of claim 15,

wherein the propagated parameter information includes at least one of dataset type, information rate or action latency information.

17. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a function descriptor object;

propagating parameter information from an input of the transport object to an output of the transport object; and

substituting an other function descriptor object in place of the function descriptor object based upon the propagated parameter information;

wherein the function descriptor object is logically equivalent to the other function descriptor object;

wherein a data set of the function descriptor object is equal to a data set of the other function descriptor object,

wherein an information rate of the function descriptor object is less than or equal to an information rate of the other function descriptor object, and

wherein an action latency of the function descriptor object is greater than or equal to action latency of the other function descriptor object.

18. (Previously Presented) The computer implemented method of claim 17, wherein the propagated parameter information includes at least one of dataset type, information rate or action latency information.

19. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a variant dataset type function descriptor object;

propagating explicit dataset type information from an input of the transport object to an output of the transport object; and

substituting at least one explicit dataset type equivalent function descriptor object in place of the variant dataset type function descriptor object based upon the propagated explicit dataset type information.

20. (Previously Presented) A computer implemented method of directing symbol substitution comprising:

associating an output of a transport object with a variant dataset type function descriptor object;

propagating explicit dataset type information from an input of the transport object to an output of the transport object; and

substituting at least one explicit dataset type equivalent function descriptor object in place of the variant dataset type function descriptor object based upon the propagated explicit dataset type information;

wherein the variant dataset type function descriptor object is logically equivalent to the at least one explicit dataset type equivalent function descriptor object.

21. (Previously Presented) A computer implemented method of defining dataset type during synthesis of a design comprising:

associating an input of a variant transport with an upstream object in the design;

associating an output of the transport with a downstream object in the design;

propagating explicit dataset type information from the upstream object to the input of the transport;

propagating explicit dataset type information from the input of the transport to an output of the transport; and

propagating the explicit dataset type information from the output of the transport to the downstream object.

22. (Previously Presented) A computer implemented method of resolving a high level design into a detailed design comprising:

associating respective output nodes of one or more variant transport objects with an equivalent function descriptor object;

associating respective information with respective input nodes of the one or more variant transport objects;

propagating the respective information from the respective input nodes of the one or more variant transport objects to respective output nodes of the one or more variant transport objects; and

substituting a less variant equivalent function descriptor object into the design in place of the variant equivalent function object based upon the respective propagated information.

23. (Previously Presented) The method of claim 22,

wherein the propagated information includes dataset type information.

24. (Previously Presented) The method of claim 22,

wherein the propagated information includes information rate information.

25. (Previously Presented) A computer implemented method of resolving a high level design into a detailed design comprising:

creating a graphical diagram in a computer system display,

which represents an algorithm

which includes a variant equivalent function descriptor graphical object

which includes one or more variant transport graphical objects, each including an input node and an output node, and

wherein the diagram represents the variant equivalent function descriptor graphical object coupled to one or more respective output nodes of one of the one or more variant transport graphical objects;

automatically creating a design in a computer readable medium,

which corresponds to the diagram,

which includes a variant equivalent function descriptor design object that corresponds to the variant equivalent function descriptor graphical object

which includes one or more variant transport design objects that correspond to the one or more variant transport graphical objects,

wherein each variant transport design object includes an input node and an output node, and

wherein the variant equivalent function descriptor design object is coupled to one or more respective output nodes of one of the one or more variant transport design objects;

associating respective information with respective input nodes of the one or more variant transport design objects;

propagating the respective information from the respective input nodes of the one or more variant transport design objects to respective output nodes of the one or more variant transport design objects coupled to the variant equivalent function descriptor design object; and

substituting a less variant equivalent function descriptor design object into the design in place of the variant equivalent function design object based upon the propagated explicit information.

26. (Previously Presented) The method of claim 25 further including:

substituting a less variant equivalent function descriptor graphical object into the diagram in place of the variant equivalent function graphical object based upon the propagated explicit information.

27. (Previously Presented) The method of claim 25,

wherein the propagated information includes at least one of dataset type, information rate or action latency.

28. (New) The computer implemented method of claim 2,

wherein associating an output of a transport object involves associating an output of a variant transport object;

wherein the propagated information includes propagating explicit data set type information; and

wherein propagating information from an input of the transport object to an output of the transport object includes substituting an explicit data set type transport object for the variant transport object.

29. (New) A computer controlled behavioral synthesis method for transforming a high level behavioral design encoded in a computer readable medium into a lower level design, comprising:

associating at least one first behavior object in the design with at least one select object in the design;

associating at least one second behavior object in the design with the at least one select object;

associating at least one third behavior object in the design with the at least one select object;

propagating control information from the third behavior object to the select object; and

associating the at least one first behavior object with the second behavior object in the design if the propagated control information has a first value; and

removing the at least one second behavior object from the design if control information having a second value is propagated to the select object.

30. (New) The method of claim 29 further including:

removing the select object from the design if control information having a first value or a second value is propagated to the select object.

31. (New) The method of claim 29 further including:

providing at least one parameter to the at least one third behavior object; and

using the parameter to transform the at least one third behavior object into control information.

32. (New) A computer controlled behavioral synthesis method for transforming a high level behavioral design encoded in a computer readable medium into a lower level design, comprising:

associating at least one first behavior object in the design with at least one select object in the design;

associating at least one second behavior object in the design with the at least one select object;

associating at least one third behavior object in the design with the at least one select object;

providing at least one parameter to the at least one third behavior object;

using the parameter to transform the at least one third behavior object into control information

propagating control information from the at least one third behavior object to the select object; and

associating the first behavior object with the at least one second behavior object in the design if the propagated control information has a first value;

removing the at least one second behavior object from the design if control information having a second value is propagated to the select object; and

removing the select object from the design if control information having a first value or a second value is propagated to the select object.

33. (New) A computer controlled behavioral synthesis method for transforming a high level behavioral design encoded in a computer readable medium into a lower level design, comprising:

associating at least one first behavior object in the design with at least one select object in the design;

associating at least one second behavior object in the design with the at least one select object;

associating at least one third behavior object in the design with the at least one select object;

associating at least one fourth behavior object in the design with the at least one select object;

propagating control information from the fourth object to the select object;

associating the first behavior object with the at least one second behavior object in the design if the propagated control information has a first value; and

associating the first behavior object with the at least one third behavior object in the design if the propagated control information has a first value.

34. (New) The method of claim 33 further including:

removing the at least one select object from the design.

35. (New) The method of claim 33 further including:

removing the third behavior object from the design if control information having the first value is propagated to the select object; and

removing the second behavior object from the design if control information having the second value is propagated to the select object.

36. (New) The method of claim 33 further including:

removing the fourth behavior object from the design.

37. (New) The method of claim 33 further including:

removing the at least one select object from the design; and

removing the fourth behavior object from the design.

38. (New) The method of claim 33 further including:

removing the at least one select object from the design;

removing the fourth behavior object from the design

removing the third behavior object from the design if control information having the first value is propagated to the select object; and

removing the second behavior object from the design if control information having the second value is propagated to the select object.

39. (New) The method of claim 33 further including:

providing at least one parameter to the fourth behavior object object; and

using the parameter to transform the fourth behavior object into control information.

40. (New) The method of claim 33 further including:

providing at least one parameter to the fourth behavior object; and

using parameter data set type to drive a propagate data set process and to drive a flatten process that transform the at least one fourth behavior object to the control information.

41. (New) The method of claim 33,

wherein associating at least one first behavior object in the design with at least one select object in the design includes connecting at least one first transport behavior object between the first behavior object and the select object;

wherein associating at least one second behavior object in the design with the at least one select object includes connecting at least one second transport behavior object between the second behavior object and the select object;

wherein associating at least one third behavior object in the design with the at least one select object includes connecting at least one third transport behavior object between the third behavior object and the select object; and

wherein associating at least one fourth behavior object in the design with the at least one select object includes connecting at least one fourth transport behavior object between the fourth behavior object and the select object.

42. (New) A computer controlled behavioral synthesis method for transforming a high level behavioral design encoded in a computer readable medium into a lower level design, comprising:

associating at least one first behavior object in the design with at least one select object in the design;

associating at least one second behavior object in the design with the at least one select object;

associating at least one third behavior object in the design with the at least one select object;

propagating control information from the at least one third behavior object to the select object; and

associating the first behavior object with the at least one second behavior object in the design if the propagated control information has a first value; and

not associating the at least one first behavior object with the at least one second behavior object in the design if control information having the first value is not propagated to the select object.

43. (New) The method of claim 42 further including:

removing the at least one select object from the design in response to propagating of the control information.

44. (New) The method of claim 42 further including:

removing the select object from the design if control information having the first value is propagated to the select object.

45. (New) The method of claim 42 further including:

removing the second behavior object from the design if control information having the first value is not propagated to the select object.

46. (New) The method of claim 42 further including:

removing the at least one select object from the design in response to propagating of the control information; and

removing the second behavior object from the design if control information having the first value is not propagated to the select object.

47. (New) The method of claim 42,

wherein not associating the first behavior object with the second behavior object in the design if control information having the first value is not propagated to the select object further includes:

not associating the first behavior object with the second behavior object in the design if control information having a second value is propagated to the select object.

48. (New) The method of claim 42,

wherein not associating the first behavior object with the second behavior object in the design if control information having the first value is not propagated to the select object further includes,

not associating the first object behavior with the second behavior object in the design if control information having a second value is propagated to the select object; and further including:

removing the second behavior object from the design if control information having the second value is propagated to the select object.

49. (New) The method of claim 42 further including:

providing at least one parameter to the third behavior object; and

using the parameter to transform the third behavior object into control information.

50. (New) The method of claim 42 further including:

providing at least one parameter to the third behavior object; and

using parameter data set type to drive a propagate data set process and to drive a flatten process that transform the at least one third behavior object to the control information.

51. (New) The method of claim 42 further including:

associating at least one fourth behavior object in the design with the third behavior object;

acquiring at least one parameter by the at least one third behavior object from the fourth behavior object; and

using the parameter to transform the at least one third behavior object into control information.

52. (New) The method of claim 42 further including:

associating at least one fourth behavior object in the design with a the third behavior object;

acquiring at least one parameter by the at least one third behavior object from the at least one fourth behavior object;

using parameter data set type to drive a propagate data set process and to drive a flatten process that transform the at least one third behavior object into one or more atomic level objects; and

using parameter value information to drive atomic level object resolution rules that transform the one or more atomic level objects into the control information.

53. (New) The method of claim 42,

wherein associating at least one first behavior object in the design with at least one select object in the design includes connecting at least one first transport behavior object between the at least one first behavior object and the select object;

wherein associating at least one second behavior object in the design with the at least one select object includes connecting at least one second transport behavior object between the second behavior object and the select object;

wherein associating at least one third behavior object in the design with the at least one select object includes connecting at least one third transport behavior object between the at least one third behavior object and the select object;

wherein associating the at least one first behavior object with the at least one second behavior object in the design if the propagated control information has a first value includes connecting at least one fourth transport behavior object between the first behavior object and the select object; and

wherein not associating the first behavior object with the second behavior object in the design if control information having the first value is not propagated to the select object includes not connecting at least one fourth transport behavior object between the at least one first behavior object and the select object.

54. (New) A computer controlled behavioral synthesis method for transforming a high level behavioral design encoded in a computer readable medium into a lower level design, comprising:

associating at least one first behavior object in the design with at least one select object in the design;

associating at least one second behavior object in the design with the at least one select object;

associating at least one third behavior object in the design with the at least one select object;

providing at least one parameter to the at least one third behavior object; and

using the parameter to transform the third behavior object into control information;

propagating control information from the at least one first behavior object to the select object; and

associating the first behavior object with the at least one second behavior object in the design if the propagated control information has a first value;

not associating the at least one first behavior object with the at least one second behavior object in the design if control information having the first value is not propagated to the select object;

removing the select object from the design if control information having the first value is propagated to the select object; and

removing the at least one second behavior object from the design if control information having the first value is not propagated to the select object.

55. (New) A method of changing a locus of a synthesis process within a design database that includes a plurality of behavior objects, comprising:

propagating explicit DS type information to a respective variant DS type input of a variant first behavior object;

substituting an explicit first equivalent function (EF) behavior object for the variant first behavior object, wherein the explicit first EF behavior object includes an explicit DS type output; and

propagating explicit DS type information from the explicit DS type output to a respective variant DS type input of a variant second object.

56. (New) The method of claim 55,

wherein propagating the explicit DS type information from the explicit DS type output to a respective variant DS type input of a variant second object includes, substituting an explicit second object for the variant second object.

57. (New) The method of claim 55,

wherein propagating explicit DS type information to a respective variant DS type input of the variant first behavior object includes, substituting an explicit third object for a variant third object; and

wherein propagating the explicit DS type information from the explicit DS type output to a respective variant DS type input of a variant second object includes, substituting an explicit second EF object for the variant second object;

58. (New) The method of claim 55,

wherein the variant second object includes a variant transport object; and

wherein propagating the explicit DS type information from the explicit DS type output to a respective variant DS type input of a variant second object includes, substituting an explicit transport object for the variant transport object.

59. (New) The method of claim 55,

wherein propagating explicit DS type information to a respective variant DS type input of the variant first behavior object includes, substituting an explicit first transport object for a variant first transport object;

wherein the variant second object includes a variant second transport object; and

wherein propagating the explicit DS type information from the explicit DS type output to a respective variant DS type input of a variant second object includes, substituting an explicit second transport object for the variant second transport object.

60. (New) A computer implemented method of resolving variant data set type into explicit data set type in a design, the method comprising:

propagating an explicit data set type within the design to an input of a variant exposer;

identifying within an object library an exposer that includes an input having a data set type that matches the propagated explicit data set type and that includes an output explicit data set type that is different from the input data set type; and

substituting the identified exposer in the design in place of the variant exposer.

61. (New) The computer implemented method of claim 60,

wherein propagating includes substituting an explicit data set type transport associated with the input of the variant exposer in place of a variant data set type transport associated with the input of the variant exposer.

62. (New) A computer implemented method of resolving a variant data set type behavioral object in a design, while maintaining unchanged a pattern of connections between behavioral objects in the design, the method comprising:

substituting a first equivalent function behavior object in the design in place of a first variant data set type behavior object;

inserting in the design a first collector that includes an input data set type that matches an output data set type of the substituted-in first equivalent function behavior object and that includes an output data set type different from an output data set type of the substituted-in first equivalent function behavior object;

propagating the output data set type of the inserted first collector within the design to an input of a variant exposer;

substituting in the design an exposer that includes an input data set type that matches the propagated output data set type of the inserted first collector and that includes an output data set type different from the propagated output data set type of the inserted first collector;

substituting a second equivalent function behavior object in the design in place of a second variant data set type behavior object and that includes an input data set type that matches the output data set type of the substituted-in exposer.

63. (New) The method of claim 62 further including:

inserting in the design a second collector that includes an input data set type that matches an output data set type of the substituted-in second equivalent function behavior object and that includes an output data set type different from an output data set type of the substituted-in second equivalent function behavior object.

64. (New) The method of claim 62 further including:

identifying within an object library an exposer that includes an input data set type that matches the propagated output data set type of the inserted first collector and that includes an output having a data set type that is different from the propagated output data set type of the inserted first collector.

65. (New) The method of claim 62 further including:

identifying within the object library a first behavioral object with a behavior that is equivalent to that of the first variant data set type object and that includes an input data set type that matches the propagated data set type and that includes an output data set type different from the propagated explicit data set type.

66. (New) The method of claim 62 further including:

identifying within the object library a first behavioral object with a behavior that is equivalent to that of the first variant data set type object and that includes an input data set type that matches the propagated data set type and that includes an output data set type different from the propagated explicit data set type; and

identifying within the object library a second behavioral object with a behavior that is equivalent to that of the second variant data set type object and that includes an input data set type that matches the output data set type of the substituted-in exposer.

67. (New) A computer implemented method of removing an exposer and a collector from a design comprising:

identifying in a design a first transport that connects to both an exposer and a collector;

determining whether an output of the collector and an input of the exposer have matching data set type;

if the data set type of the collector output matches the data set type of the exposer input, then substituting in place of the first transport and the collector and the exposer, a second transport and a third transport;

such that the second transport is connected in the design to a first input transport that had been connected in the design to an input of the collector and to a first output transport that had been connected to an output of the exposer; and

such that the third transport is connected in the design to a second input transport that had been connected in the design to another input of the collector and to a second output transport that had been connected to an output of the exposer.

68. (New) A computer implemented method of converting a data set of a first type to a data set type of a second type comprising:

casting up a first data set of a first type to a prescribed data set type that is large enough to encompass a data set of a second type; and

casting down the casted up first data set from the prescribed data set type to the second data set of the second data set type.

69. (New) The computer implemented method of claim 68,

wherein casting up includes increasing a number of bits in the first data set such that an increased number of bits in the first data set matches a number of bits in the prescribed data set type; and

wherein casting down includes reducing a number of bits in the casted up first data set such that a reduced number of bits in the first set matches a number of bits in the second data set type.

70. (New) The computer implemented method of claim 68,

wherein casting up includes adding a number of bits in the first data set such that a number of bits in the added to first data set matches a number of bits in the prescribed data set type, and further including the step of,

assigning each of the added bits a prescribed value; and

wherein casting down includes eliminating a number of the added bits from the casted up first data set such that a number of bits in the eliminated from first data set matches a number of bits in the second data set type.